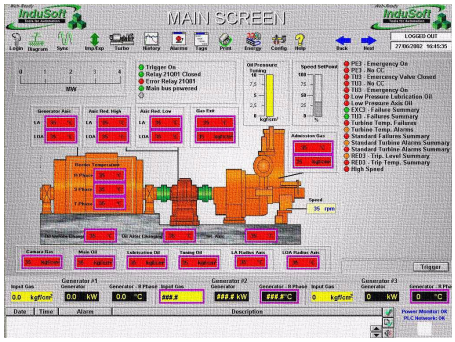
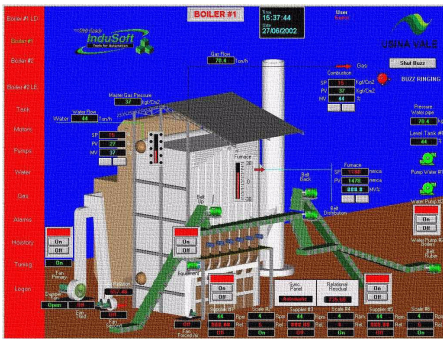
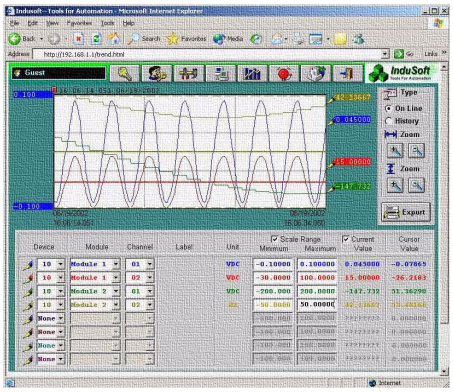
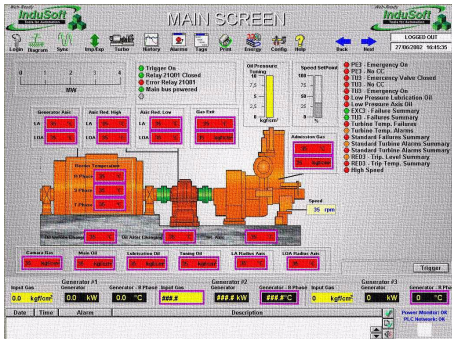


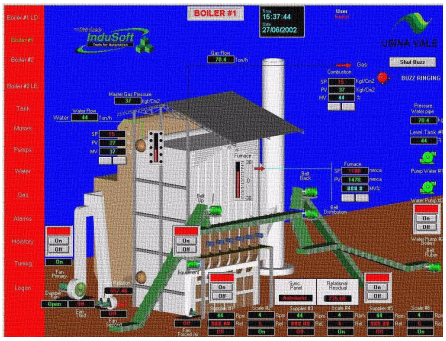


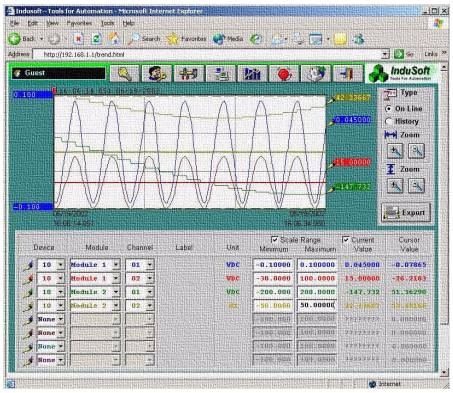


InduSoft Web Studio Getting Started Guide

www.InduSoft.com
877-INDUSOFT







InduSoft Web Studio Getting Started Guide

www.InduSoft.com
877-INDUSOFT

Windows, Windows XP, Windows 2000, and Windows NT are registered trademarks of Microsoft Corp. in the United States and other countries.

Other brand or product names are trademarks or registered trademarks of their respective owners.

Copyright © 2003 InduSoft, Ltd. All rights reserved worldwide.
This document shall not be reproduced or copied in any manner without expressed written authorization from InduSoft.

The information contained within this document is subject to change without notice. InduSoft, Ltd. does not guarantee the accuracy of the information.

PN: IND-GSG-WD002

Contents

Chapter 1. Welcome to InduSoft Web Studio 1

Overview 1
About InduSoft Web Studio 2
Main Features 3
System Requirements 5

Chapter 2. About this Publication 7

Text Conventions 7
Mouse and Selection Conventions 8
Windows Conventions 8
Related Publications 10

Chapter 3. Installing the Software 11

Installing InduSoft Web Studio 11
Starting InduSoft Web Studio 13
Installing CEView Software 13
Uninstalling InduSoft Web Studio 16

Chapter 4. Working with the IWS Interface 19

Overview 19
Using the Title Bar 21
Reading the Status Bar 21
Using the Menu Bar 22
Using the Toolbars 24
Using the Workspace 26

Chapter 5. Working with Tags 29

Naming Tags and Tag Fields 29
Tag Syntax 29
Tag Field Syntax 30
Working with Tag Folders 32
Understanding the Tag Types 34
Using Array Tags 34
Using Indirect Tags 35

Chapter 6. Creating an IWS Application 37

Creating a New Application 37
Specifying Project Settings 39
Creating Tags 40
Creating the Start-up Screen (`main.scr`) 41

Creating the Synoptic Screen (`synoptic.scr`)....45
Creating a Screen Title....45
Creating Date / Time Text Objects....46
Creating an Exit Icon....47
Testing the Application....48
Creating a Tank Object....49
Creating a Level Slider Object....52
Selecting a Tank....54
Testing the Application....55
Configuring the I/O Communication Driver....56

Chapter 7. Running the Web-Based Application....63

Chapter 8. Running Applications from a Remote Station....67

Chapter 1. Welcome to *InduSoft Web Studio*

InduSoft Web Studio™ (or *IWS*) is a powerful, integrated tool that exploits key features of Microsoft® Windows® NT/2000/XP and Windows® CE, and enables you to build full-featured *SCADA* (Supervisory Control And Data Acquisition) or *HMI* (Human-Machine Interface) applications for your Industrial Automation business.

Overview

This *InduSoft Web Studio Getting Started Guide* is intended for individuals using InduSoft Web Studio for the first time. This publication will help you quickly familiarize yourself with the basic functions of InduSoft Web Studio.

> **Note:**
> InduSoft assumes you are familiar with the Windows NT/2000/XP operating system. However, if you need assistance as you work, we suggest using the **Help** feature on the Windows desktop **Start** menu.

This Getting Started guide is divided into the following chapters:

- *This chapter* provides a brief overview of the InduSoft Web Studio product, including its main features and system requirements.
- **Chapter 2, About this Publication**: Provides information about this publication; such as conventions used and related publications.
- **Chapter 3, Installing the Software**: Explains how to install InduSoft Web Studio and CEView on your system.
- **Chapter 4, Working with the IWS Interface**: Describes all the elements of the IWS development environment; including menubars, toolbars, dialogs, buttons, menus, and so forth.
- **Chapter 5, Working with Tags**: Describes the procedures for creating and editing tags for the tags database; including a description of the proper syntax, and tag types.
- **Chapter 6, Creating an IWS Application**: Explains how to create and edit applications, objects, screens, and drivers. Also explains the procedure for testing your applications.
- **Chapter 7, Running the Web-Based Application**: Explains how to start and run your Web-based applications.
- **Chapter 8, Running Applications from a Remote Station**: Explains how to run and manage applications remotely.

About InduSoft Web Studio

InduSoft Web Studio applications consist of animated operator-interface screens, communication drivers (for PLCs and I/O protocols such as Modbus, DFS, Profibus, and so forth), tags database, and additional modules such as alarm monitors, logic, trend charts, recipes, schedulers, and security system.

IWS applications interface with industrial I/O systems and other Windows applications in the run-time environment using ODBC, DDE, NetDDE, OPC, or TCP/IP protocols. You can also operate the application through a standard Web browser (Internet Explorer or Netscape).

The product consists of two parts:

- The *development system* software, which runs on a desktop, laptop, or industrial PC running Windows NT/2000/XP.
- The *run-time system* software, which runs on any operator interface workstation running Windows NT/2000/XP or Windows CE.

> **Note:**
> The run-time system software for the Windows CE operating system (*CEViеw*) is usually pre-loaded on the HMI. If necessary, you can update the CEView version of the development system software by downloading the current version to the HMI or PDA.

You can run IWS applications on your development workstation or download it to a run-time station (through a serial or TCP/IP connection) and run the application using the IWS or CEView run-time software. The workstation processes scan data from connected devices according to parameters defined in the application and then react to, display, store, and upload the data.

Main Features

The InduSoft Web Studio product offers the following features and functions:

- Integrated Windows development environment with toolbars, dialogs, and menus:
  - Shortcut menus, which can be accessed by right-clicking on any area of the development environment (options vary according to context)
  - Customizable fly-over toolbars
  - Tasks, objects, and controls organized in a tree-view explorer
- Full-featured objects and dynamics (the ability to modify object properties, execute commands, or inset values to tags used to build screens on the fly at runtime):
  - Configurable objects such as buttons, rectangles, ellipse, polygons, lines, and text
  - Dynamic properties such as bar graphs, color, resizing, position, hide/unhide, rotation, command, hyperlink, and text input/output
  - Online and historical alarm list displays
  - Online and historical trending
  - Alignment and distribution tools
  - Background bitmap layer creation and editing
  - Graphics importation
  - Active-X object containers
- Online remote management and configuration
- Microsoft DNA architecture compliance, with full OPC and XML support
- Web interface enabled, which exports application screens to a “thin” client through the Internet/intranet and by exchanging data online through the TCP/IP protocol
- Symbol library with more than 100 symbols and dynamic objects, such as pushbuttons, meters, sliders, switches, text and numeric displays, LED-style indicators, pipes, bumps, icons, vehicles, valves, frames, motors, gauges, and common controls
- Debugging tools:
  - *Database Spy* window to monitor/force tag values and execute functions
  - *LogWin* module to record OPC, DDE, and TCP/IP transactions, modules activation, trace tags, and so forth
  - Cross-referencing to locate tags throughout the project
  - Online system and network diagnostics

- Powerful and flexible *Tags Database* (Boolean, Integer, Real, and String tags), array tags, classes, and indirect tag-pointers
- Open architecture with API exchanges and tag values with external software
- Translation editor, which enables you to translate an application into several different languages, and switch between them while the runtime system is online
- TCP/IP client and server modules to exchange tag values and configure redundancy systems
- More than 200 drivers for different devices (such as PLC) from several manufacturers; such as Allen-Bradley, Siemens, GE-Fanuc, as well as standard protocols such as MODBUS RTU/ASCII, DeviceNet, Profibus, Interbus, and so forth
- OPC Server and OPC Client with integrated OPC Browser
- Screen and object password-protected runtime security (256 levels)
- Logical expressions and a scripting language with more than 200 functions
- Recipe and Report (ASCII, UNICODE, and RTF formats) builders integrated into the product
- Event scheduler based on date, time, or data condition (100ms resolution)
- Multi-layer application, which means modular worksheets and screens can be merged easily to other applications
- Full integration with PC-based control programs (imports tag databases) such as ISaGRAF, SteepleChase, Think&Do, and ASAP
- Dial-Up functions to trigger, monitor, and hang-up a dial-up connection with the RAS Server of remote stations
- Functions to send e-mail from IWS (or CEView)
- Real-time project documentation
- Screen resolution converter

Note:

IWS provides different product types for each level of application responsibility. However, IWS does not support some features in certain product types (such as CEView). You can review the **`TargetVersions.pdf`** document on the InduSoft Web Studio CD-ROM for detailed information about the limitations of each product-type limitations.

System Requirements

To develop and run applications with IWS, you must install the following hardware and software:

- IBM-compatible computer with an Intel® Pentium II-compatible processor or higher
- Windows NT/2000/XP operating system for development
- Windows NT/2000/XP or Windows CEv3.00 or Windows CE.NET operating system for runtime
- Minimum of 128MB random-access memory (*RAM*); *256MB or higher recommended*
- MS Internet Explorer 6.0 or higher
- Minimum of 150MB free hard disk space (required for the program without any application files); *300MB is recommended*
- CD-ROM drive (can be on a different computer)
- Standard keyboard with function keys F1 through F12
- Parallel printer port (*optional*)
- 100% IBM-compatible VGA or SVGA display adapter with 32MB Video RAM (*VRAM*) (*optional for runtime when running Web-based applications*)
- Microsoft-compatible pointing device (such as a mouse, trackball, or touch-screen)
- One or two COM ports and adapters for downloading applications (*optional*)
- Ethernet connection for downloading applications (*optional*)

> **Note:**
> IWS is UNICODE-compliant and will not run on a non-UNICODE-compliant operating system (such as Windows 9x/ME). You can, however, run the Web Thin Clients of IWS applications on non-UNICODE operating systems.

Chapter 2. About this Publication

This chapter provides the following information about this *InduSoft Web Studio Getting Started Guide* and other InduSoft publications:

- **Text Conventions**: Describes the text-formatting conventions used in this publication to help you read and assimilate information quickly.
- **Mouse and Selection Conventions**: Describes the conventions used in this publication for using a mouse and selecting objects in a Windows environment.
- **Windows Conventions**: Describes conventions used in this publication for working in a Windows environment.
- **Related Publications**: Lists other InduSoft publications (available on the IWS installation CD-ROM) you can read for more information about the InduSoft Web Studio product.

Text Conventions

This publication uses special formatting to help you quickly identify certain items, as follows:

- Titles, labels, and new terms are indicated using *italic* text.
- File names, messages, and screen text are indicated using bold, monospaced text (for example, **`D:\Setup.exe`**).
- Variables and information you must provide are indicated using bold, italicized monospaced text enclosed in **<** and **>** brackets (for example, Run the **`<driver name>.exe`** file indicates you must provide a driver name.)
- Buttons, menu options, and keyboard keys are indicated in **bold**.
- Text requiring special emphasis is in ***bold italic***.
- Some information is segregated into **Tip**, **Note**, and **Caution** boxes as follows:
  - **Tips** provide useful information to save development time or to improve application performance.
  - **Notes** provide supplemental information related to the surrounding text, usually the text just preceding the note.
  - **Cautions** provide information necessary to prevent damage or problems when running the application.

Mouse and Selection Conventions

Because most PCs used for application development run a version of Windows with a mouse, this publication assumes you are using a mouse. This publication also assumes that your mouse is configured so that the left button is the primary button and the right button is the secondary button.

This publication uses the following mouse and selection conventions:

- **Double-click** means to quickly click twice on an object with the left mouse button.
- **Right-click** means to click once on an object with the right mouse button.
- **Click** and **Select** both mean to click once on an object with the left mouse button. In general, you *click* on buttons and *select* from lists.
- *Select* also means to use your pointing device to highlight or specify an item on the computer screen. Selecting an object with a touch-screen is usually the same as selecting with a mouse, except that you use your finger to touch (select) a screen object or section.
- You can also use keys on your keyboard to select objects/options. For example, you can use the **Tab** key to move between options, the **Enter** key to open menus, and the **Ctrl** or **Alt** keys in combination with a letter key (for example, **Ctrl+S**) to select a menu option having an underlined letter (**Save**).
- **Drag** means using your mouse/cursor to move an object to a new location on your computer screen. Position your cursor over the object and press the mouse button. As you move the mouse, you move the object. (Usually an outline of the object will move along with the cursor.) When you have the object positioned where you want it, release the mouse button.

Windows Conventions

This publication uses the following Windows conventions:

- **Dialog boxes** (or *dialogs*) are windows that allow you to enter information.
- **Text boxes** (or *fields*) are areas in dialogs where you can type in text.
- **Radio buttons** are white circles in which a black dot ( ) appears or disappears when you click on the button. Typically, the black dot indicates that the option or function is enabled (*active*). The absence of a black dot indicates the option or function is disabled (*inactive*).
- **Check-boxes** are white squares in which a check ( ) appears or disappears when you click on it with the cursor. Typically, the check indicates that the option or function is enabled (*active*). The absence of a check indicates the option or function is disabled (*inactive*).

- **Buttons** are icons in boxes that appear "pressed" when you click on them.
- **Lists** are panes (white boxes) in windows or dialog boxes containing two or more selectable options.
- **Combo-boxes** have arrows that, when clicked, show part or all of an otherwise concealed list.

Using a Combo-Box

- **Interface** or **Development Environment** refers to the entire InduSoft Web Studio window.
- **Dockable windows** are windows that you can drag to an edge of the interface and merge with that edge.
- **Toolbars** are dockable windows that contain only buttons and text boxes.

> **Note:**
> The dialogs and procedures described in this publication are valid for Windows 2000. Some terms may vary according to the operating system (type, language, and version) you are using.

Related Publications

You may want to review the following publications for additional information about InduSoft Web Studio:

- *InduSoft Web Studio Users Guide and Technical Reference Manual*: Provides detailed information about using InduSoft Web Studio and explains all its features and functions.
- *InduSoft Web Studio Training Manual*: Describes how to build an application, step-by-step, with the main product features. You can use this document as a self-training manual.
- *Drivers User Guides*: Explain how to configure individual InduSoft drivers, according to their unique protocol characteristics. One customized user guide is included with each InduSoft driver.

> **Note:**
> These publications are located in the *Documentation* folder on the *InduSoft Web Studio* CD-ROM. IWS installs the Driver User Guides in the *DRV* subdirectory of the *InduSoft Web Studio* folder. Technical information also is available from the **Help** menu on the IWS menu bar.

Chapter 3. Installing the Software

This chapter provides instructions for installing, starting, and uninstalling InduSoft Web Studio and CEView. The information is organized as follows:

- **Installing InduSoft Web Studio:** Explains how to install InduSoft Web Studio on your computer.
- **Starting InduSoft Web Studio:** Explains how to run InduSoft Web Studio.
- **Installing CEView Software:** Explains how to install CEView on your computer.
- **Uninstalling InduSoft Web Studio:** Explains how to uninstall InduSoft Web Studio and CEView.

Installing InduSoft Web Studio

InduSoft Web Studio (*IWS*) provides development tools for all InduSoft applications, and it runs on the Microsoft Windows NT/2000/XP operating systems.

You can install IWS from the InduSoft Web Studio installation CD-ROM or download the installation files from the InduSoft Web site at (http://www.indusoft.com). For Windows CE applications, you use IWS to download CEView (run-time software) to the Windows CE HMI using a serial or TCP/IP link.

The IWS installation program automatically creates necessary directories, copies files to your hard drive, and creates the IWS icons in your desktop folder.

Notes:

- You must have Administrator privileges on a Windows NT/2000/XP workstation to install InduSoft Web Studio.
- You must uninstall all previous versions of InduSoft Web Studio or install the newer version in a different directory. Also, you cannot install the same version of IWS in two different paths of the same computer.

Use the following procedure to install IWS from the CD-ROM:

1. Turn on the power to your development computer (Windows NT, 2000, or XP) and be sure that no other programs are running.
2. Insert the installation CD-ROM into the CD-ROM driver.

> **Notes:**
> A CD Browser window should display automatically. If not, you can start the program manually from Windows Explorer. Navigate to the **`D:\Installation`** directory (where *D* is your CD-ROM drive), and run the **`Setup.exe`** file.

3. Double-click on the *Installation* folder, and then double-click on the **InduSoft Web Studio** icon to launch the *InstallShield® Wizard*.
4. Follow the instructions provided by the *InstallShield Wizard* to proceed with the installation.
5. When prompted to restart Windows, click the **Yes, I want to restart my computer now** radio button, then click **OK**.

> **Notes:**
> Refer to the *Product Licensing - User Guide* manual (available on you IWS CD-ROM) for instructions about licensing IWS and/or CEView.

Starting InduSoft Web Studio

To run IWS, double-click the InduSoft Web Studio shortcut on the desktop or select **Start** → **Programs** → **InduSoft Web Studio** → **InduSoft Web Studio**.

Starting IWS

> ⇨ **Tip**:
> You can run the IWS development environment under any video setting; however, InduSoft recommends using a resolution of 800x600 (or higher) with more than 256 colors for a more pleasing environment. Application resolution (screen size) is independent of the operating system resolution.

Installing CEView Software

When installing IWS under Windows NT/2000/XP the CEView runtime files are stored in the following folder:

<*InduSoft Web Studio Folder***>\Redist\<***WinCE version***>\<***Processor Type***>**

where:

- ***<InduSoft Web Studio Folder>*** is the installation directory chosen during installation (**C:\Program Files\InduSoft Web Studio** is the default installation directory).
- ***<Processor Type>*** is the processor platform. InduSoft provides a CEView runtime for all processor platforms supported by the WinCE operating system (Arm, Mips, MipsFP, Pocket-Arm, Pocket-Mips, Pocket-SH3, PPC, SH3, SH4, Thumb, and x86).
- ***<WinCE version>*** is the Windows CE version (for example, WinCE300, WinCE400, and so forth).

To install CEView, use the following steps:

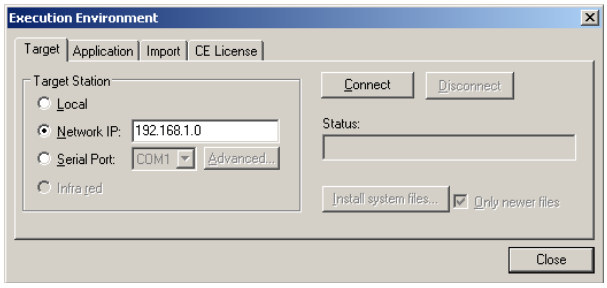
1. Power-on the Windows CE device, and the *Remote Agent* dialog should launch automatically. If not, copy the **CESERVER.EXE** file from the **\<*InduSoft Web Studio Folder*>\Redist\<*WinCE version*>\<*Processor Type*>\BIN** directory of the Windows NT/2000/XP computer where you installed IWS.
2. Paste the file into the **\<*non-volatile*>** folder of your WinCE device and run it.

   > **Note:**
   > There are different ways to copy a file into a WinCE device (for example, you can map a shared folder from the Windows NT/2000/XP computer in the WinCE device or using *ActiveSync*). If you need assistance copying this file into the WinCE device, contact InduSoft technical support or the hardware manufacturer.

   After executing the **CESERVER.EXE** file, the *Remote Agent* dialog launches in the WinCE device.
3. Click the **Setup** button in the *Remote Agent* dialog and configure the link (serial or TCP/IP) to connect the WinCE device to the Windows NT/2000/XP computer.

   > **Note:**
   > For better performance, InduSoft recommends using a TCP/IP link between the WinCE device and the Windows NT/2000/XP computer to download and upload files.
4. Start IWS on the Windows NT/2000/XP computer.
5. Select **Project** → **Execution Environment** from the main menu bar.

6. When the *Execution Environment* dialog displays, select a connection type (**Network IP** or **Serial Port**) and configure its settings (for example, **COM Port** or **IP Address**).



Execution Environment Dialog

7. Click the **Connect** button to connect IWS to the WinCE device.

8. Click the **Install System Files** button from the *Execution Environment* window (*Target* tab) to download the CEView files to the WinCE device.

Uninstalling InduSoft Web Studio

If you must remove IWS from your system, follow these instructions:

> **Caution**:
>
> Before uninstalling IWS, be sure to back-up any files you might need later into the …**`\ InduSoft Web Studio\Projects\`** folder. Also, be sure you have the current (or newest) version of the IWS installation CD-ROM or diskettes so you can re-install the software again.
>
> *You will lose the product softkey license when you uninstall IWS.*

1. From the *Windows* taskbar, select **Start** → **Settings** → **Control Panel** to open the Control Panel.

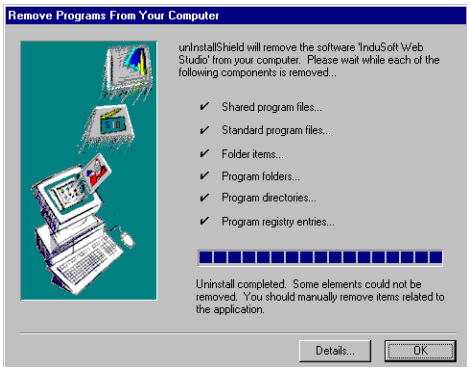
Opening the Control Panel

2. Double-click on the **Add/Remove Programs** icon ( ) in the *Control Panel* window.

3. When the *Add/Remove Programs Properties* dialog displays (see figure), select **InduSoft Web Studio** from the list and click the **Add/Remove**... button.

4. When the *Confirm File Deletion* dialog displays, click the **Yes** button.

The *Uninstall Shield Wizard* and the *Remove Programs from Your Computer* dialogs display.


Remove Programs From Your Computer
unInstallShield will remove the software 'InduSoft Web Studio' from your computer. Please wait while each of the following components is removed...
Shared program files...
Standard program files...
Folder items...
Program folders...
Program directories...
Program registry entries...
Uninstall completed. Some elements could not be removed. You should manually remove items related to the application.
Details...
OK

Removing the Program

5. When the `Uninstall successfully completed` message displays and the **OK** button becomes active, click **OK**. Verify that IWS is no longer listed in the *Add/Remove Programs Properties* dialog.

6. Click the **Cancel** button or the **close** button ( ), to close the *Add/Remove Programs Properties* dialog, and then close the Control Panel.

7. Open the *Windows Explorer* and browse to the directory containing the *InduSoft Web Studio* directory.

8. Verify that all of the IWS files and folders were deleted. (You can manually delete any that remain.)

> **Note:**
> The uninstall tool should not delete any files that you created or modified in the …**`\ InduSoft Web Studio\Projects\`** folder.

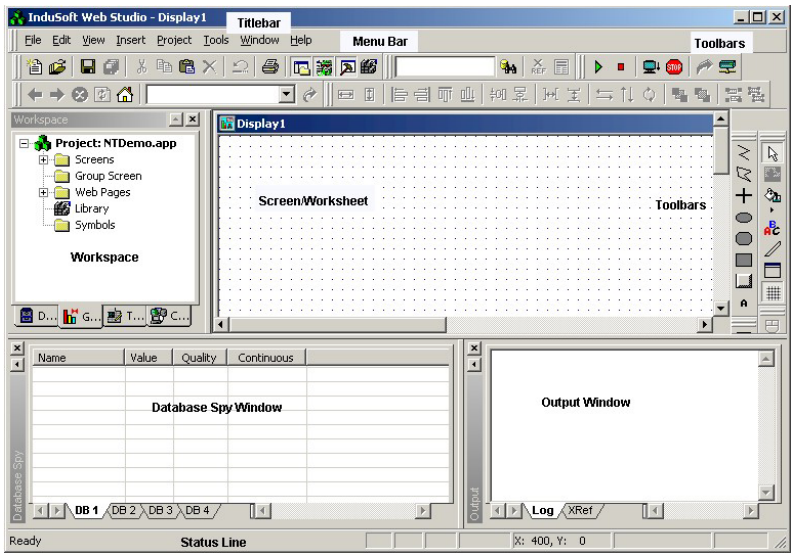
Chapter 4. Working with the IWS Interface

This chapter describes the InduSoft Web Studio *development environment* (interface). The information in this chapter is organized as follows:

- **Overview**: Provides a brief description of the tools and interfaces that make up the IWS development environment.
- **Title Bar:** Describes the title bar.
- **Status Bar:** Describes the status bar.
- **Menu Bar:** Describes the different menu options that you can access from the IWS main menu bar.
- **Toolbars:** Describes the IWS toolbars.
- **Workspace:** Describes the IWS Workspace.

Overview

InduSoft Web Studio uses standard, Windows-like tools and interfaces to make the product user-friendly. IWS also provides an integrated, unique development environment (see figure) for easy access to tools and information.



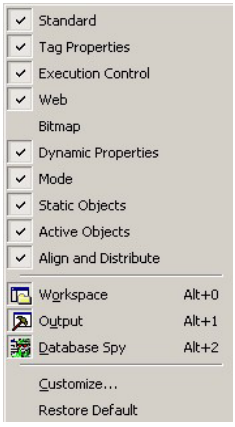
InduSoft Web Studio Development Environment

The development environment consists of the following basic areas:

- **Title Bar**: Indicates the active screen or worksheet
- **Status Bar**: Provides quick access to actual information
- **Menu Bar**: Contains the main product options and controls, which you can easily access using the cursor or your keyboard keys
- **Auxiliary Toolbars**: Provide shortcuts to the main commands used in the development environment
- **Displays Building Toolbars**: Contain features and tools used to create or edit objects and dynamics in the application screens
- **Workspace**: Provides tree-view control from which you can access project worksheets and screens
- **Database Spy Window**: Provides a debugging tool, which you can use to monitor and force tags and to execute functions
- **Output Window**: Displays debugging messages
- **Displays/Worksheets**: Provides an area where you can edit screens and worksheets

Note:
The preceding figure shows the development environment areas and windows in their default position. You can customize this environment as needed by changing the position of the areas.

You can *right*-click the mouse almost anywhere inside the development environment to display a pop-up menu (similar to the following figure), relating to the context of where you clicked.



Sample IWS Pop-Up Menu

Using the Title Bar

The title bar (located along the top of the IWS window) displays the **InduSoft Web Studio** icon, the product name, and the name of the active, open screen or worksheet (if any).

Typical IWS Title Bar

The title bar also contains the following three buttons (from left to right):

- **Minimize** button (): Click this button to minimize the IWS window.
- **Resize/Maximize** button (/): Click the button to toggle between the two options,
  - **Resize** tiles the IWS window
  - **Maximize** maximizes the IWS window to fill your computer screen
- **Exit** (or **Close**) button (): Click this button to automatically save the database then close IWS. If you modified any screens or worksheets, IWS prompts you to save your work. This button function is similar to the **Exit** command on the **File** menu.

> **Note:**
> Closing the development environment does not close the IWS run-time tasks. To close your run-time tasks, click the **Stop** icon on the *Execution* toolbar or select **Project** → **Stop Application** option from the main menu bar.

Reading the Status Bar

The status bar (located along the bottom of the IWS window) contains fields used to identify toolbar buttons and provide information about the active screen (if any).

Sample Status Bar

The fields are as follows (from left to right):

- **Hint** field: Provides a short description of any toolbar button or display object touched by the cursor.
- **Caps Lock** field: Indicates whether the keyboard **Caps Lock** key is on (***CAP***) or off (*empty*).
- **Num Lock** field: Indicates whether the keyboard **Num Lock** key is on (***NUM***) or off (*empty*).

- **Scroll Lock** field: Indicates whether the keyboard **Scroll Lock** key is on (***SCRL***) or off (*empty*).
- **ID** field: Displays the ID number of a selected screen object.
- **Screen Coordinate** field: Displays the current location of the cursor (or pointer) on the active screen. When you select a screen object, this field displays the object's coordinates in the top-left corner. Where: ***X*** is the number of pixels from the left edge of the screen and ***Y*** is the number of pixels from the top of the screen.
- **Object Size** field: Displays the size (in pixels) of a selected object, where ***W*** is the width and ***H*** is the height.
- **No DRAG** field: Indicates whether dragging is disabled (***No DRAG***) or enabled (*empty*) in the active screen.

  You might want to disable dragging to change the object properties of an object without moving it from its current location.

> ⇨ **Tip:**
>
> You can enable/disable dragging by pressing **Ctrl+D**. Also, you can click on any object and use the arrow keys on the keyboard to move objects on the screen, pixel by pixel.

Using the Menu Bar

The menu bar contains the following menus:

Main Menu Bar

- **File**: Contains options that enable you to create, open, close, save, and print application projects and files, and allows you to close the IWS development environment.
- **Edit**: Contains options that enable you to edit your displays and worksheets by cutting, copying, pasting, formatting, and setting security levels. You can also undo edits, delete, search and replace elements in these displays/worksheets.

- **View**: Contains options that enable you to manage which tools and toolbars are visible in the development environment. This menu also provides shortcuts to the dialogs you open most frequently, enables you to customize your toolbars, restore defaults, open the libraries, zoom/unzoom, and set screen attributes.
- **Insert**: Contains options that enable you to create and configure a variety of application tags, tag classes, documents, drivers, users, security settings, screens, and ActiveX objects.
- **Project**: Contains options to execute applications locally and remotely, and provides links used to configure general application settings.
- **Tools**: Contains options that provide links to auxiliary tools.
- **Window**: Contains options that enable you to manage open displays and worksheets.
- **Help**: Contains options that link to information about IWS and InduSoft.

> **Note:**
> - The menu bar is dockable.
> - If you right-click on the menu bar, a pop-up menu displays. From this pop-up, you can restore the default location of the menu bar and toolbars.
> - You also can use this pop-up to hide toolbars and windows and to customize the development environment.

Using the Toolbars

IWS provides several toolbars containing icons (shortcuts) that enable you to perform different actions within the program. This section describes the function and default location of each toolbar.

> **Notes:**
>
> All toolbars are dockable screen objects. You can move a toolbar to a different screen location by clicking on its title bar and dragging it to a new location.
>
> For more information about the toolbars discussed in this section, see the *InduSoft Web Studio Users Guide and Technical Reference Manual*.

The following toolbars contain general-purpose tools, and they are located across the top of the workspace, just below the menu bar by default:

- **Standard** toolbar: Allows you to perform general actions, such as file management and printing functions.

Standard Toolbar

- **Tag Properties** toolbar: Allows you to create, locate, and access different tags and their properties.

Tag Properties Toolbar

- **Execution Control** toolbar: Allows you to execute and manage an application locally or from a remote location.

Execution Control Toolbar

- **Web** toolbar: Allows you to open HTML files.

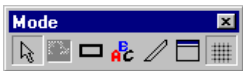
Web Toolbar

- **Align and Distribute** toolbar: Allows you to edit screen objects.

Align and Distribute Toolbar

The following toolbars contain screen-editing tools. These toolbars are located along the right side of the interface window by default and they are enabled only while you are editing graphic screens:

- **Mode** toolbar: Allows you to edit your screens.



Mode Toolbar

- **Bitmap** toolbar: Allows you to access the bitmap screen editor tools. (This toolbar is available only when the *Background Picture* layer is active. You enable the *Background Picture* layer in the *Screen Attributes* dialog.)

Bitmap Toolbar

> **Note:**
> The *Bitmap* toolbar is hidden by default.

- **Static Objects** toolbar: Allows you to create polygons, rectangles, lines, and other objects for your screen.

Static Objects Toolbar

- **Dynamic Properties** toolbar: Allows you to apply dynamics to objects or a group of objects. Dynamics enable you to modify object properties on the fly (during runtime) according to tag values. Some dynamics also enable you to execute commands or insert values (set points) to the tags.

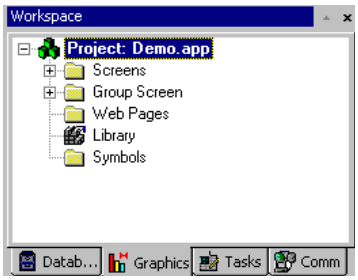
Dynamic Properties Toolbar

- **Active Objects** toolbar: Allows you to create dynamic objects. Active objects typically require more parameters than static objects and provide embedded dynamics.

Active Objects Toolbar

Using the Workspace

The IWS *Workspace* window is a user-friendly interface that enables you to quickly find and/or create any application component (tags, screens, worksheets, and so forth). Application components are organized in a tree-view with each one having its own icon and customized description. You can move, resize, or hide the *Workspace* window.



The IWS Workspace

The *Workspace* window is divided into four tabs, as follows:

- **Database** tab: Provides access to all tags in the application and security system components. This tab includes the following folders:
  - *Application Tags*
  - *Classes*
  - *Shared Database*
  - *Internal Tags*
  - *Security*
- **Graphics** tab: Provides access to all screens and symbols in the application. This tab includes the following folders and icons:
  - *Screens*
  - *Group Screen*
  - *Web Pages*
  - *Library*
  - *Symbols*
- **Tasks** tab: Provides access to all task worksheets in the application. This tab includes the following folders:
  - *Alarms*
  - *Trend*
  - *Recipes*
  - *Report*
  - *ODBC*
  - *Math*
  - *Scheduler*
- **Comm** tab: Provides access to all worksheets configured to establish communication with another device or software using available protocols. This tab includes the following folders:
  - *Drivers*
  - *OPC*
  - *TCP/IP*
  - *DDE*

> **Note:**
> You can right-click on all folders and components to open a menu relating to that folder or component.

Chapter 5. Working with Tags

This chapter provides information needed to create and edit tags, including:

- **Naming Tags and Tag Fields:** Explains the required syntax for naming tags and tag fields.
- **Working with Tag Folders:** Explains the purpose of tag folders.
- **Understanding the Tag Types:** Describes the different tag types.
- **Using Array Tags:** Explains how to use array tags.
- **Using Indirect Tags:** Explains how to use indirect tags.

Naming Tags and Tag Fields

This section provides guidelines for naming *tags* (database location identifiers) and *tag fields* (a set of parameters inherent to each tag in the database). Applications use tag fields during runtime.

Tag Syntax

You must observe the following syntax guidelines when naming a tag:

- Use letters, numbers, and the underscore (_) character.
- *Do not* use the following characters:

  ` ~ ! @ # $ % ^ & * ( ) - = \ + \ [ ] { } < > ?

- Tag names must begin with a letter.
- Maximum tag length is 32 characters.
  Maximum class member length is 16 characters.
- Tag names must be unique—do not specify the same name for two different tags, unless you are creating an array tag and specify a unique index (see “Using Array Tags”).
- Tag names are not case-sensitive (for readability however, we recommend using uppercase and lowercase characters. For example, use *TankLevel* instead of *tanklevel*).
- Tag names must be different from internal tag names and math functions.

  Some valid tag names include:

  – *Temperature*
  – *pressure1*
  – *count*
  – *x*

Tag Field Syntax

You can use the *Tag Properties* dialog (click the **Tag Properties** icon located on the *Tag Properties* toolbar) to specify tag field parameters. To access a tag field you must use the following syntax:

<*TagName*>-><*TagField*> (for example, **second->Max**)

You can access the following tag fields during runtime:

Tag Field Name	Description of Value Associated with Each Field	Tag Type Associated with Field				R=Read Only RW=Read+Write
		Boolean	Integer	Real	String	
Description	Description of tag configured in Tags database.	✓	✓	✓	✓	RW
Max	Maximum value that can be written to the tag at runtime.	✗	✓	✓	✗	RW
Min	Minimum value that can be written to the tag at runtime	✗	✓	✓	✗	RW
Quality	Tag quality (192=GOOD; 0=BAD). Updates every time tag receives the result of an expression or a value from a communication task (Driver or OPC). Invalid expressions (such as division by 0) or reading communication errors associated with tag, sets quality to BAD.	✓	✓	✓	✓	R
Size	Array Size. If the tag is not an array tag, returns the value 0.	✓	✓	✓	✓	R
TimeStamp	Records time and date when a tag changes value.	✓	✓	✓	✓	R
Unit	Brief description (up to 9 characters) of an engineering unit (such as Kg) for a tag value	✓	✓	✓	✓	RW
B0 … B31	Value (0 or 1) of any of the 32 bits (b0, b1, b2, … b31) of an integer tag. (B0: LSB B31: MSB).	✗	✓	✗	✗	RW

Tag Field Name	Description of Value Associated with Each Field	Tag Type Associated with Field				R=Read Only RW=Read+Write
		Boolean	Integer	Real	String	
AlrStatus	Status (integer value) of any currently active alarms associated with a tag. Each bit of the integer value indicates a specific status: • Bit 0 (LSB): HiHi Alarm active • Bit 1: Hi Alarm active • Bit 2: Lo Alarm active • Bit 3: LoLo Alarm active • Bit 4: Rate Alarm active • Bit 5: Deviation+ Alarm active • Bit 6: Deviation- Alarm active For example: • If **Tag->AlrStatus** = 2, "Hi" alarm is active. • If **Tag->AlrStatus** = 3, "HiHi" and "Hi" alarms are active simultaneously. • If **Tag->AlrStatus** = 0, there are no active alarms For Boolean tags, only 1 (bit 1), 4 (bit 2) or 16 (bit 4) values are returned.	✓	✓	✓	✘	R
Ack	Specifies whether alarms associated with the tag require acknowledgement. This field has only two possible values: • **0**: Alarms do not require acknowledgment • **1**: At least one alarm requires acknowledgment	✓	✓	✓	✘	RW
AlrDisable	Specifies whether alarms associated with a tag will become active. This field has only two possible values: • **0**: Enables alarm and when an alarm condition occurs, the alarm becomes active. • **1**: Disables alarm so that even if an alarm condition occurs, the alarm will not become active.	✓	✓	✓	✘	RW
HiHi	• If **0**, HiHi alarm is inactive. • If **1**, HiHi alarm is active.	✘	✓	✓	✘	R
Hi	• If **0**, Hi alarm is inactive. • If **1**, Hi alarm is active.	✓	✓	✓	✘	R

Tag Field Name	Description of Value Associated with Each Field	Tag Type Associated with Field				R=Read Only RW=Read+Write
		Boolean	Integer	Real	String	
Lo	• If **0**, Lo alarm is inactive. • If **1**, the Lo alarm is active.	✓	✓	✓	✗	R
LoLo	• If **0**, LoLo alarm is inactive. • If **1**, the LoLo alarm is active.	✗	✓	✓	✗	R
Rate	• If **0**, Rate alarm is inactive. • If **1**, the Rate alarm is active.	✓	✓	✓	✗	R
Devp	• If **0**, Dev+ alarm is inactive. • If **1**, the Dev+ alarm is active.	✗	✓	✓	✗	R
Devm	• If **0**, Dev- alarm is inactive. • If **1**, Dev- alarm is active.	✗	✓	✓	✗	R
HiHiLimit	Limit value for HiHi alarm.	✗	✓	✓	✗	RW
HiLimit	Limit value for Hi alarm.	✗	✓	✓	✗	RW
LoLimit	Limit value for Lo alarm.	✗	✓	✓	✗	RW
LoLoLimit	Limit value for LoLo alarm.	✗	✓	✓	✗	RW
RateLimit	Limit value for Rate alarm.	✗	✓	✓	✗	RW
DevSetpoint	Set point value for Deviation alarms.	✗	✓	✓	✗	RW
DevpLimit	Limit value for Deviation+ alarm.	✗	✓	✓	✗	RW
DevmLimit	Limit value for Deviation- alarm.	✗	✓	✓	✗	RW

Note:
If the application tries writing a value outside the range specified in the **Min** and **Max** fields, the Tags Database will not accept the new value and writes a warning message in the *LogWin*. If you configure both **Min** and **Max** properties with the value `0` (zero), any value applied to the tag type can be written to the tag.

Caution:
You cannot use tag fields (such as Bit fields) to configure *Alarm* or *Trend* worksheets.

Working with Tag Folders

You can use tags as communication points with field equipment, results of calculations, alarm points, and so forth. In IWS, all tags are organized into folders on the *Database* tab according to their origin (application, internal, or shared). IWS also provides a folder for compound-tags, named *classes*.

The following is a description of the different IWS tag folders:

- **Application Tags**: User-defined tags created for screens, to read from and write to field equipment, for control, auxiliary tags to perform mathematical calculations, and so forth.

- **Internal Tags**: Tags predefined by IWS. Internal tags have predetermined functions (such as time, date, acknowledge alarms, storage of the logged-on user name and so forth). You cannot delete or modify these tags, but you can access their values from any IWS task.
- **Shared Tags**: Tags created in PC-based control software and imported into the IWS environment. You cannot edit shared tags in the IWS environment, but you can modify these tags using PC-based control software and then re-import the modified tags to update the IWS database. Consequently, you can configure shared tags for any IWS task just as any other tag.
- **Classes**: Structures that allow for high-level encapsulation in the application database. When you create a class-type tag, the tag contains a whole set of values rather than a single value. You create classes by grouping elements, called *members*. The maximum number of members for any class depends on the product specification.

Understanding the Tag Types

A tag can be one of the following types:

- **Boolean**: Boolean or digital variable (0 or 1).
- **Integer**: Integer number (positive, negative, or zero), equivalent to C-type long integer (4 bytes). For example: 0, 5, -200.
- **Real**: Real number (float) internally stored as a double word, equivalent to C-type double (8 bytes, for example: 2.12, -10.5).
- **String**: Character string up to 256 characters that contains letters, numbers, or special characters. For example: Recipe product X123, 01/01/90, *** On ***.
- **Class**: User-defined, compound tag.

The preceding icons (and their respective tag types) are located in folders on the *Database* tab.

Using Array Tags

IWS tags can consist of a single value or an array of values.

An *array tag* is a set of tags that all have the same name, but use unique indexes to differentiate between each tag (a matrix of *n* lines and *1* column). The maximum array size permitted will depend on the product specification.

You can use the following syntax to access an array tag:

<***ArrayTagName***>**[*ArrayIndex*]**

For example: **tank[1]**, **tank[2]**, **tank[3]**, and **tank[500]**.

> **Caution:**
>
> You can specify the maximum index for each array tag in the **Size** column of any datasheet. Specify size *n* to indicate that the array tag has positions from 0 to *n*.
>
> For example, if the size of TagA is *3*, the tag positions could be: **TagA[0]**, **TagA[1]**, **TagA[2]**, and **TagA[3]**.

In many cases, using array tags will simplify the configuration task. For example, if you want a display to monitor each tank, you could use array tags to configure a single display containing tags linked to any tank (using the **tk** tag as an index containing the tank number). For example,

pressure[tk], **temperature[tk]**, and **temperature[tk +1]**.

An array index can be a tag, a numeric value, or an expression with the arithmetic operator **+**.

> **Note:**
> When referring to an array having an index with the arithmetic operation **+**, you must use the following syntax:
>
> ***<ArrayTagName>*[*<NumValue1>* + *<NumValue2>*]**
>
> Where ***<NumValue1>*** and <***NumValue2***> can be integer tags or numerical constants. For example:
> **temperature[tk+2], temperature[tk+6],** or
> **temperature[TagA+ TagB]**

Using array tags in an IWS task can save a significant amount of application development time. For example, if you wanted tag points related to the temperature of four tanks, the conventional configuration method is:

- **temperature1 high temperature on tank 1**
- **temperature2 high temperature on tank 2**
- **temperature3 high temperature on tank 3**
- **temperature4 high temperature on tank 4**

Using array tags simplifies this task, as follows:

temperature[j] high temperature on tank {j}

> **Note:**
> When you create an ***n***-position array tag, the system creates ***n*+1** positions (from **0** to ***n***). For example: **tag_example[15]** with **Array Size=4** has five elements, **(start position=0, end position=4)**.

Using Indirect Tags

IWS supports indirect access to tags in the database. For example, consider a tag *X* of the string type. This tag can hold the name of any other tag in the

database (in other words, this tag can provide a pointer to any other tag type, including the class type). The syntax for an indirect tag is straightforward:

@<***IndirectTagName***>

For example, assume that the ***X*** tag holds the **TEMP** string. Reading and/or writing to **@*X*** provides access to the value of the **TEMP** tag.

> **Note:**
> Any tag created as a string type is a potential indirect tag (pointer).

Chapter 6. Creating an IWS Application

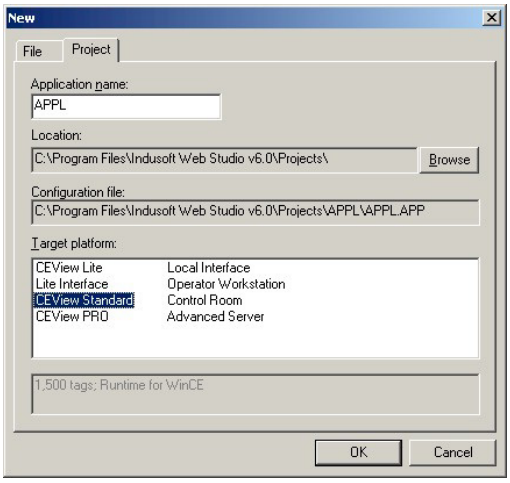
This chapter explains (using a step-by-step tutorial) how to create a working IWS application and how to install and configure an I/O driver.

- **Creating a New Application:** Explains how to create IWS applications.
- **Specifying Project Settings:** Explains how to specify various settings for your application.
- **Creating Tags**: Explains how to create new tags and add them to the IWS Tags database.
- **Creating the Start-up Screen (main.scr):** Explains how to create the main start-up screen.
- **Creating the Synoptic Screen (synoptic.scr):** Explains how to create a synoptic screen for your application.
- **Configuring the I/O Communication Driver**: Explains how to configure an I/O communication driver for your application.

Creating a New Application

Use the following procedure to create a new IWS application:

1. Select **File** → **New** from the InduSoft Web Studio main menu bar.
2. When the *New* dialog opens, click on the *Project* tab.



Select the Project Tab on the New Dialog

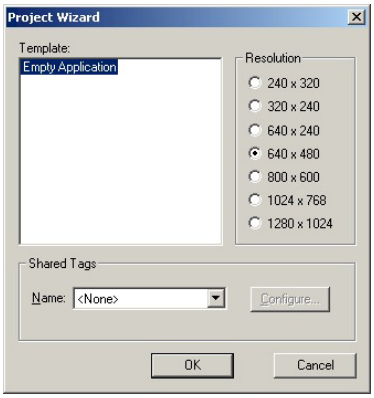
3. Type the name of your application into the **Application name** text box (for this example, type `GetStart`).

   IWS automatically creates a new directory of the same name and assigns your application file to that directory (notice the **Configuration file** text box in the previous figure). To put your application file somewhere other than in the *\Projects* subfolder of *InduSoft Web Studio*, click **Browse** and navigate to the preferred directory location.

4. Select a platform from the *Target Platform* list (for this example, select **CEView Standard**), then click **OK** to proceed.

> **Note:**
> You use CEView Lite and CEView Standard target platforms for WindowsCE-compliant run-time applications.

5. When the *Project Wizard* dialog displays, select **Empty Application** from the *Template* list, click the **640 X 480** radio button to specify the application resolution, and then click **OK**.



Specifying an Empty Application with 640X480 Resolution

The new application file name displays in the *Workspace* window. (For this example, the file name is *Project: GetStart.APP*).

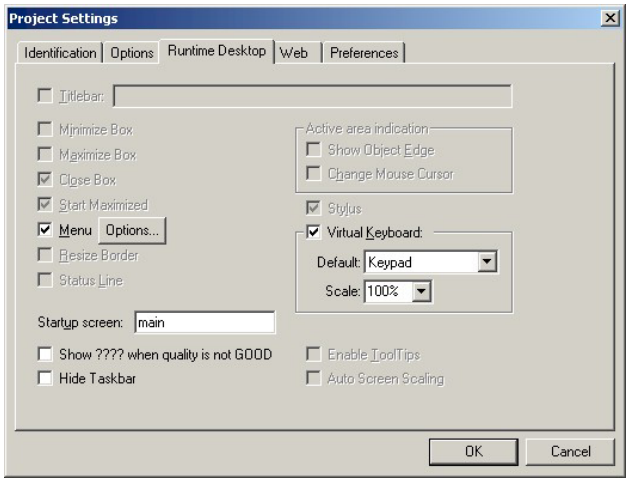
Specifying Project Settings

You can use tabs on the *Project Settings* dialog to apply certain parameters to the entire project. For example:

- Use the **Identification** tab to provide information that identifies the project application (such as project description, revision number, Company name, Author's name, field equipment, and general notes).
- Use the **Options** tab to specify automatic translations, alarm history, pc-based controls, target systems, communication drivers, OPCs, and TCP ports.
- Use the **Runtime Desktop** tab to enable/disable the run-time desktop parameters.
- Use the **Web** tab to specify, data server IP addresses, send periods (in milliseconds), URL addresses, tooltips and file compression, logging, and IP security.
- Use the **Preferences** tab to enable/disable warning messages before downloading screens to the target system.

Specify the following settings for your new *GetStart.APP* application:

1. Select **Project** → **Settings** from the main menu bar.
2. When the *Project Settings* dialog displays (see the following figure), click the *Runtime Desktop* tab and type **`main`** in the **Startup screen** field.

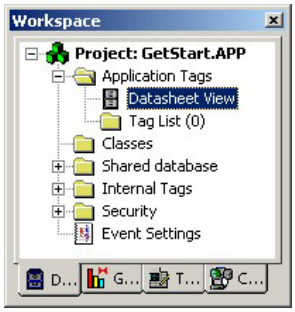


Specifying the `main` Screen

When you execute the application, IWS will open the **`main`** screen (or whatever screen you specify) automatically.

Creating Tags

A *tag* is any variable that holds a value. You can create tags at any time during the development of an application. All tags created in an application are stored in the *Application Tags* folder located on the *Database* tab in the *Workspace*.

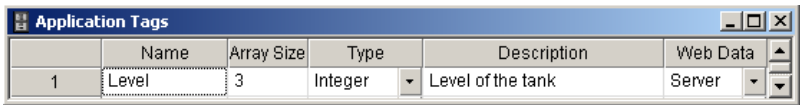


Application Tags Folder

Use the following procedure to create a new tag:

1. Double-click on the **Datasheet View** icon to open the *Application Tags* worksheet.
2. When creating a new tag, you must set the following main properties:
   - **Name**: Specify a unique tag name. (All tags must have a unique name.)
   - **Array Size**: Specify the number of elements for an array tag. (For simple tags, you must specify zero.)
   - **Type**: Specify the data type to be supported by the tag (Boolean, Integer, Real, String, or Class).
   - **Description** (*optional*): Type a description of the tag for documentation purposes only.
   - **Web Data**: Specify the communication behavior of the tag between the Server and the Web Thin Client stations.
     - Specify **Local** if you want the tag to have independent values in the Server and Web Thin Client stations.
     - Specify **Server** if you want the tag to share the same value in the Server and Web Thin Client stations.

3. Use the following parameters to create a tag for the sample application:



Notice that **Array Size** = **3** was specified for each tag. Each array element relates to one of the three tanks:

`Level[1]` → Level of the Tank #1
`Level[2]` → Level of the Tank #2
`Level[3]` → Level of the Tank #3

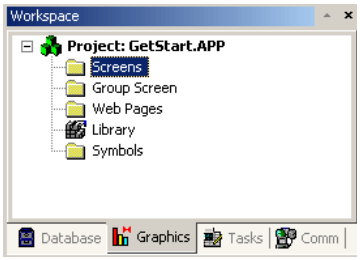
Do not use the **0** element **`(Level[0])`** in this example (although it is a valid tag).

> **Note:**
> You will create additional tags as you configure the application.

Creating the Start-up Screen (`main.scr`)

To create a new screen for your application:

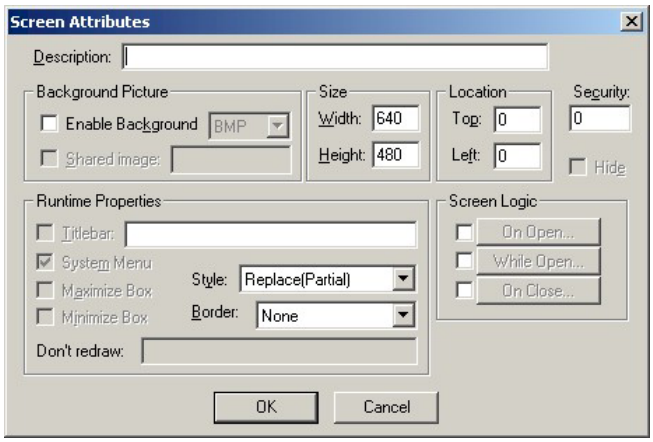
1. In the *Workspace* window, click the *Graphics* tab then right-click on the *Screens* folder.



Right-Click the Screens Folder

IWS stores all screens created for an application in this *Screens* folder.

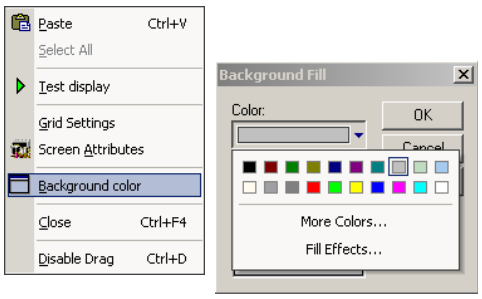
2. When the pop-up menu displays, select the **Insert** option to open the *Screen Attributes* dialog.



Screen Attributes Dialog

3. Use this dialog to set screen properties such as size and type. For this example, just click the **OK** button to accept the default settings.

4. Right-click on the new screen and select **Background color** from the pop-up menu. Select the *gray* color and click **OK** to apply that color to the screen.



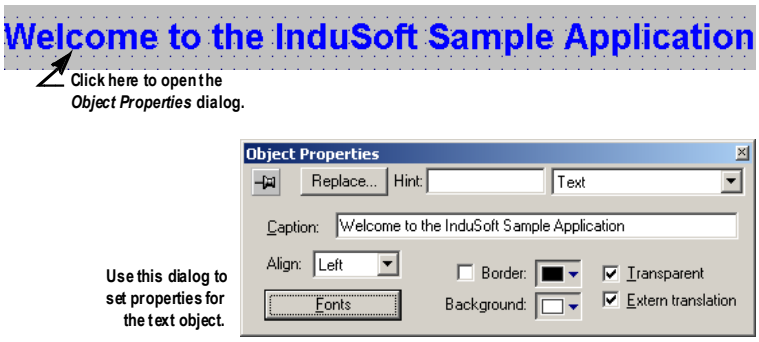
Open the Background Color Dialog and Select a Gray Color

5. Create a Text object, by clicking on the **Text** icon .

6. Click on the screen and type the following text:

   `Welcome to the InduSoft Sample Application.`

7. Double-click on the Text object to open the *Object Properties* dialog.



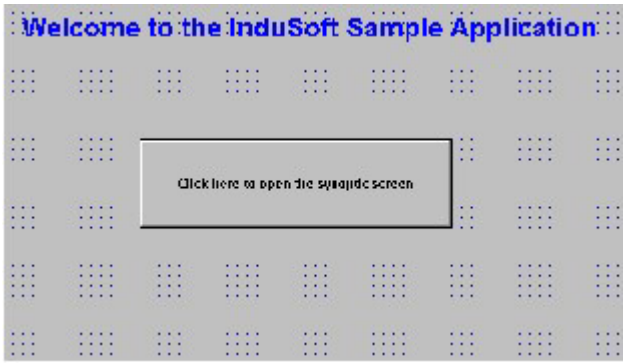
Click on the Text Object

> **Notes:**
> Double-clicking on any object opens an *Object Properties* dialog containing parameters related to that object. The features on the dialog change depending on the item selected from the combo-box The combo-box list changes depending on which properties you applied to the object.
>
> The *Object Properties* dialog contains a button that controls whether this dialog remains open. The button changes state (and function) each time you click on it, as follows:
> - Click to close the dialog when you click anywhere in the display except on an object.
> - Click to keep the dialog open regardless of where you click.
>
> Also, you can click to close the *Object Properties* dialog.

8. Check (*enable*) the **Transparent** check-box to set the background color of the Text object to transparent.
9. Click the **Fonts** button to specify font settings (for example: Type=Arial, Size=20, Color=Blue).
10. Click the **Button** icon to create a Button object.
11. Click on the screen, press the mouse button and drag the cursor to define the object size.

12. Double-click on the new Button object to open the *Object Properties* dialog and type the following text into the **Caption** field:

    `Click here to open the synoptic screen.`

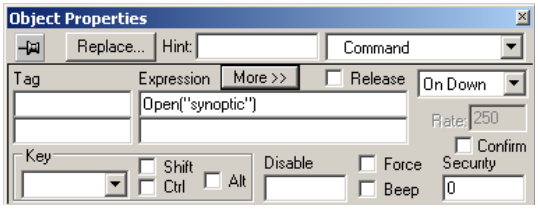


Adding a Caption to the Button

13. Select the Button object and click on the **Command** icon to apply this dynamic to the object.

14. Double-click on the Button object to reopen the *Object Properties* dialog.

15. Type `Open("synoptic")` into the *Expression* field to apply this Command dynamic to the button:



Applying the Open("synoptic") Command Dynamic

When a user clicks on this button during runtime, IWS will open the *synoptic* screen.

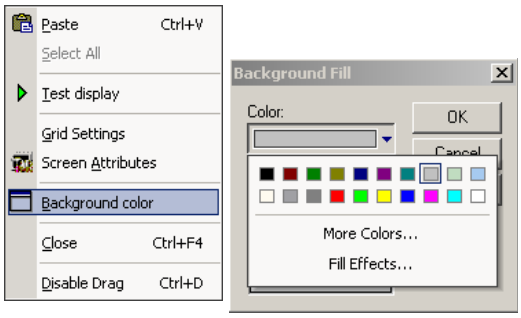
16. Finally, select **File** → **Save** from the main menu bar, and save the screen as `main.scr`.

17. To close the **main** screen, right-click on the screen and select the **Close** option from the pop-up menu.

Creating the Synoptic Screen (`synoptic.scr`)

To create a new synoptic screen, use the following steps:

1. Select the *Graphics* tab then right-click on the *Screens* folder.
2. Select the **Insert** option from the pop-up menu and the *Screen Attributes* dialog displays automatically.
3. Use the parameters on this dialog to set the main properties of the screen (size, type, and so forth). For this example, just click **OK** to accept the default settings and open the screen in the screen editor window.
4. Right-click on the screen and select **Background color**.



Selecting a Background Color

5. When the *Background Fill* dialog displays, select the gray color and click **OK** to apply this color to the screen.

Creating a Screen Title

To create a screen title Text object, use the following steps:

1. Click the **Text** icon , click on the screen, and type: Synoptic Screen.
2. Double-click on the Text object to open the *Object Properties* dialog.
3. When the dialog displays, check the transparent check-box and set the Text object's background color to transparent.
4. Click the **Fonts** button and modify the font settings (for example: Type=Arial, Size=20, Color=Blue).
5. Move the object to the top of the screen.

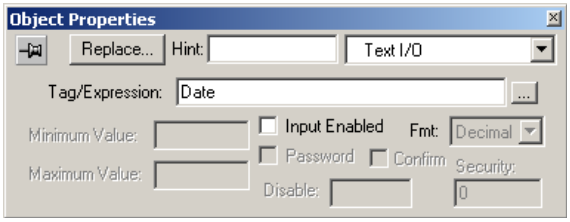
Finished Screen Title

Creating Date / Time Text Objects

> **Note:**
> Date and Time are internal tags that hold the current date and current time (respectively) from the local station. The Date and Time tags are available for any application.

To create Date and Time Text objects, use the following steps:

1. Click on the **Text** icon, then click on the screen and type: `Date: ##########`.
2. Double-click on the Text object to open the *Object Properties* dialog.
3. When the dialog opens, enable the **Transparent** check-box and set the Text object's background color to transparent.

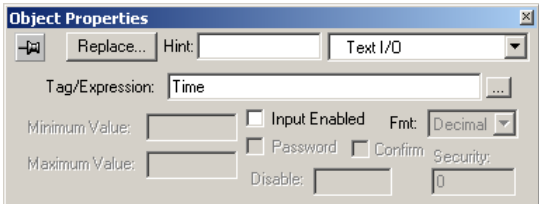


4. Select the Text object and click on the **Text I/O** icon to apply this dynamic to the object.
5. Double-click on the Text object to open the *Object Properties* dialog again.
6. Type `Date` into the *Tag/Expression* field (as shown in the following figure) to configure the Text I/O dynamic.

   During the runtime, IWS replaces the ########## chars you configured for the Text object with the value of the Date tag.
7. Click on the **Text** icon again to create a time Text object.
8. Click on the screen and type `Time: ########`.
9. Double-click on the Text object to open the *Object Properties* dialog.
10. Enable the **Transparent** check-box and set the background color to transparent.

11. Select the Text object and click on the **Text I/O** icon to apply this dynamic to the object.

12. Double-click on the Text object to reopen the *Object Properties* dialog and type `Time` into the *Tag/Expression* field (see figure) to specify the internal tag for the Text I/O dynamic.



During runtime, IWS replaces the `########` chars with the `Time` tag value.

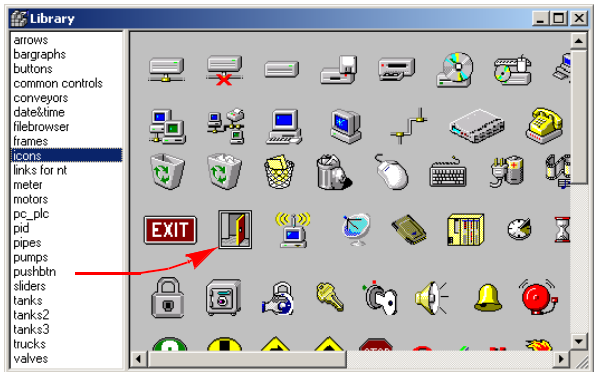
This figure illustrates how your screen should look after creating the Date and Time text objects.

Finished Date and Time Objects

Creating an Exit Icon

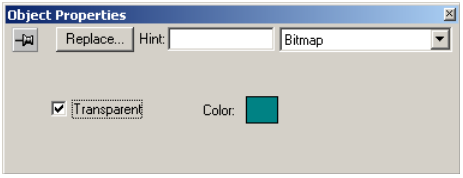
To create an **Exit** icon for the screen, use the following steps:

1. Click the **Library** icon to open the symbols library.
2. Select **icons** from the list then double-click on the **open door** icon to open the object on your screen.



Select the Open Door Icon

3. Double-click on the Bitmap object to open its *Object Properties* dialog.



4. Select the Bitmap object and click on the **Command** icon to apply this dynamic to the object.

5. Double-click on the Bitmap object to reopen the *Object Properties* dialog and type `Shutdown()` into the Expression field to configure this function for the Command dynamic and apply it to the bitmap.

   The following figure illustrates how your screen should look after creating the **Exit** icon (right side of screen). Now, when a user clicks on this icon during runtime, IWS will `Shutdown()` the application (runtime modules).

Exit Icon

Testing the Application

Use the following procedure to test your application:

1. Click the **Run application** icon (located on the IWS toolbar) to execute the application.

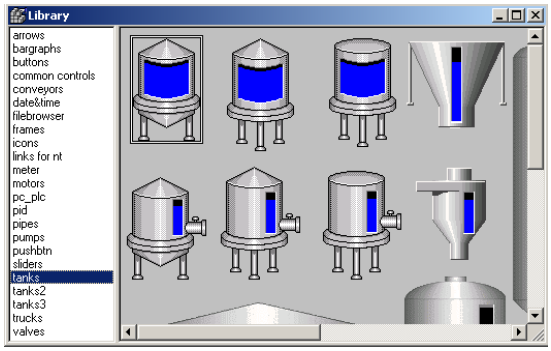
2. Click the **Exit** icon (open door bitmap) or switch back to the development environment of IWS (**Alt+TAB**) and click the **Stop application** icon on the IWS toolbar to close the application.

Creating a Tank Object

To add a Tank object to your screen, use the following steps:

1. Click the **Library** icon located on the IWS toolbar) to open the symbols library.

2. Select **tanks** from the list, and double-click on any tank to include it on the screen:



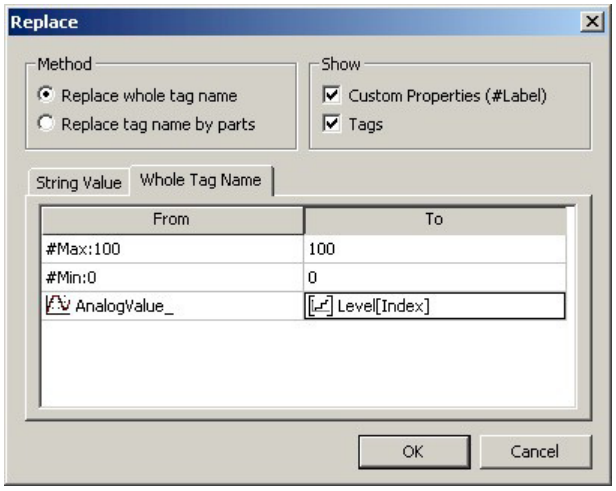
Select a Tank Object

3. Double-click on the tank to open its *Object Properties* dialog.

   A tank is a combination of different objects and dynamics from IWS (for example a rectangle, a bar graph, and so forth). You can modify the properties of this object by selecting the object or dynamic from the Group of Symbols combo-box. For this example however, you will modify just the tag associated with the tank object.

4. Click on the **Replace** button to associate a tag with the tank object.

5. When the *Replace* dialog displays (see following figure), you are going to replace the *Current Tag* **AnalogValue_** (an internal tag used to simulate level) by typing `Level[Index]` in the topmost *New Tag* field.



Replace Dialog

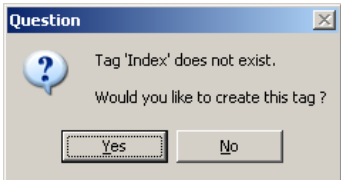
You can use the `[Index]` tag to set the array position of the `Level` tag, and show the level for any of the three tanks in the same object. For example:

When:

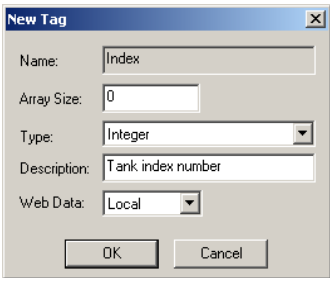
- **Index=1**, the object shows the level of Tank #1 `(Level[1])`
- **Index=2**, the object shows the level of Tank #2 `(Level[2])`
- **Index=3**, the object shows the level of Tank #3 `(Level[3])`

6. Click **OK** to confirm the tag replacement.

   Because you have not previously created an `Index` tag in the Application Tags database, IWS displays the following message:



7. Click **Yes** to create the Index tag from the screen editor. (Note that you do not have to open the application Tags database again to create tags as you configure the application).

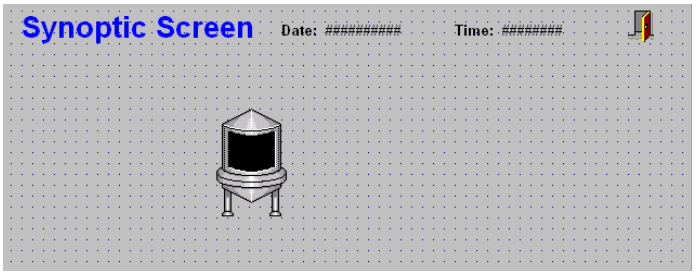


New Tag Dialog

8. You must configure the Index tag **Type** as **Integer**, **Array Size** as **0**, and **Web Data** as **Local**.

   Because the tag is **Local**, it can have different values for the Server station and the Web Thin Client station at same time. Consequently, the local user (Server station) can be monitoring the level of one tank (for example: Index=1 Tank#1) while the remote user (Web Thin Client station) is monitoring the level of another tank (for example: Index=2 Tank#2).

   The following figure illustrates how your screen should look after creating the tank object (bottom center).



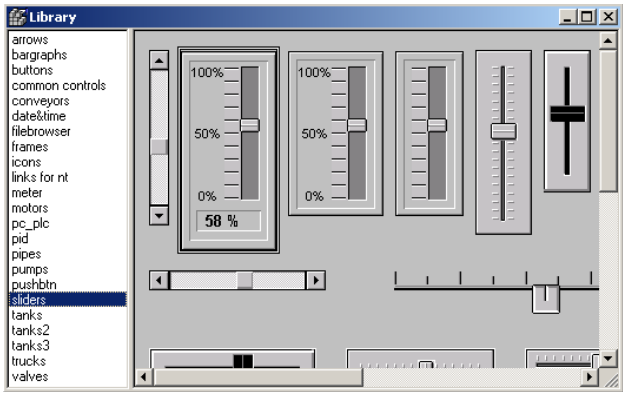
Screen with New Tank Object

Creating a Level Slider Object

To create a level slider object for your application, use the following steps:

1. Click the **Library** icon to open the symbols library.

2. Select **sliders** from the list, then double-click on any slider object in the display (see following figure) to add it to the screen.



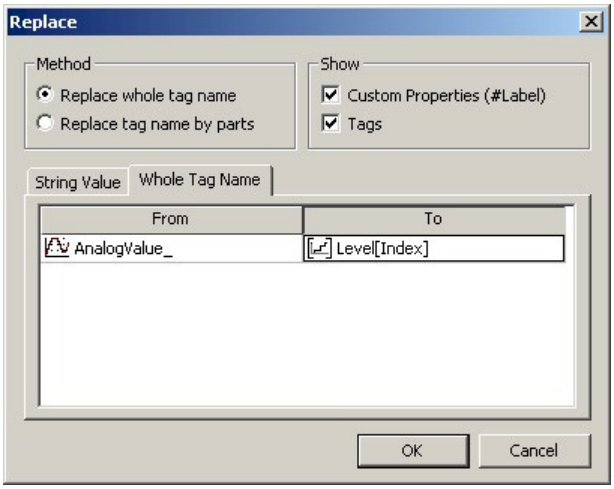
Selecting a Slider Object

3. Double-click on the slider to open its *Object Properties* dialog.

   A slider is a group of different objects and dynamics from IWS (rectangle, position, and so forth). You can modify the properties of this object by selecting the object or dynamic from the Group of Symbols combo-box. For this example, you will modify just the tag associated to the object.

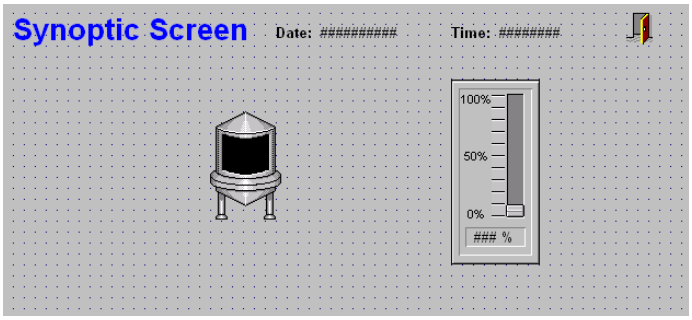
4. Click the **Replace** button to associate a tag to the object.

5. When the *Replace* dialog displays, type `Level[Index]` in the topmost *New Tag* field then click **OK** to confirm the tag replacement



Replacing the Current Tag

The following figure illustrates how your screen should look after creating the slider object (bottom right).

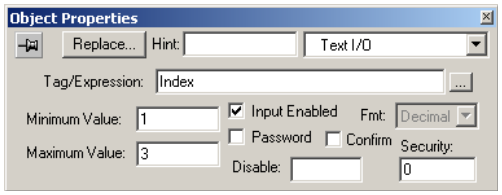


Screen with the New Slider

Selecting a Tank

To select a tank, use the following steps:

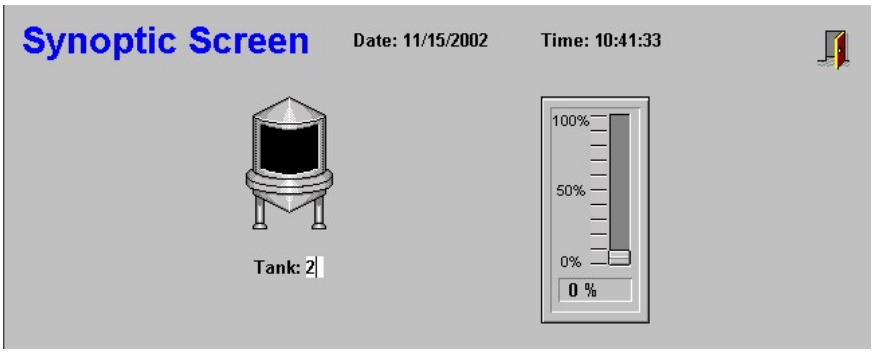
1. Click on the **Text** icon to create a Text object.
2. Click on the screen and type: `Tank: #`.
3. Double-click on the Text object to open the *Object Properties* dialog.
4. Enable the **Transparent** check-box and set the background color of the Text object to transparent.
5. Select the Text object and click on the **Text I/O** icon to apply this dynamic to the object.



6. Double-click on the Text object to reopen the *Object Properties* dialog and type `Index` in the **Tag/Expression** field to specify this internal tag for the Text I/O dynamic.
7. Enable the **Input Enabled** check-box to permit entering a new value for the Index tag during runtime.
8. Type `1` in *Minimum Value* field and `3` in the *Maximum Value* field.

   During runtime, IWS will replace the **#** characters configured in the Text object with the Index tag value.
9. Finally, select **File** → **Save** from the main menu bar, and save the screen as `synoptic.scr`.

The following figure illustrates how your screen should look when you are finished creating your application.



Finished Application Screen

10. Right-click on the screen, and select **Close** from the pop-up menu to close the main screen.

Testing the Application

To test your application, use the following steps:

1. Click on the **Run application** icon to execute the application.
2. Type the tank number (`1`, `2`, or `3`) in the **Tank** label and modify the level of each tank using the slider. Note that you can monitor/set the level of each tank independently.
3. Click the **Exit** icon (upper right corner) or switch back to the IWS development environment (**Alt**+**TAB**) and click the **Stop application** icon to close the application.

Configuring the I/O Communication Driver

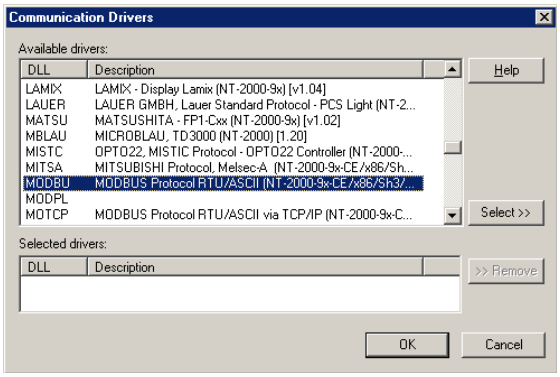
The IWS run-time software uses I/O drivers to acquire necessary data values from PLCs and other I/O devices. This section explains how to select, install, and configure an I/O driver.

1. In the *Workspace*, select the *Comm* tab and right-click the *Drivers* folder.

2. When the pop-up menu displays, select the **Add/Remove drivers** option.

Add/Remove Drivers

The *Communication Drivers* dialog displays:



Communication Drivers Dialog

3. Select a driver from the *Available drivers* list (for this example, use the **MODBU-Modbus Protocol RTU/ASCII** driver) then click the **Select** button (or double-click on the driver).

   The program moves the selected driver to the *Selected drivers* list.

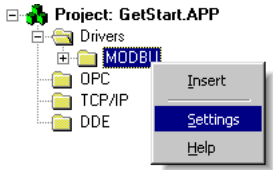
4. Click **OK** to close the window and save your changes.

> **Note:**
> To remove a driver from *Selected drivers*, select it and click **Remove**.

5. A boxed plus sign displays beside the *Drivers* folder. Click the plus sign or double-click on *Drivers* (the word or folder) to display the subfolders

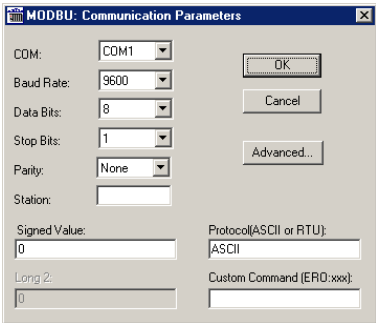
for any drivers you added. (For this example, a *MODBU* subfolder displays.)

6. Right-click on the *MODBU* folder and select **Settings** from the pop-up menu.



Select Settings from the Pop-Up Menu

7. When the *Communication Parameters* dialog displays, configure the settings to match the communication parameters configured for the PLC.

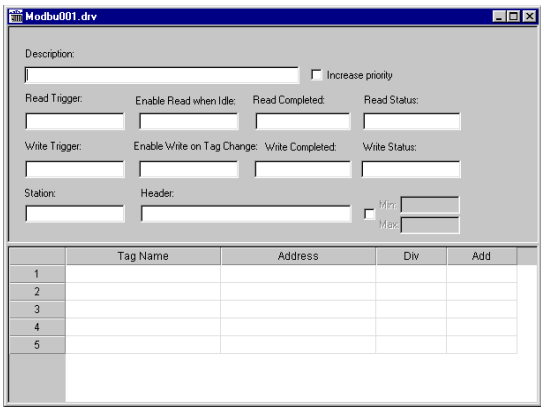


Communications Parameters Dialog for MODBU Driver

> **Tip:**
> Select **Help** → **Communication Drivers** and *MODBU Driver User Manual* for detailed information about configuring these communication parameters.

8. Right-click the *MODBU* folder again and select **Insert** from the menu.

A *Modbu001.drv* dialog displays in the *Workspace*. Notice that the dialog is divided into two sections:



Modbu001.drv Dialog

- **Header**: A gray header with several fields
- **Body**: Form of the worksheet

9. In the header section, configure these parameters as follows:
   - **Description** text box: Type `Example Modbus`. (This parameter is just for documentation and it does not affect the communication in any way.)
   - **Enable Read When Idle** text box (*Boolean values, `1` = yes and `0` = no*): Type `1` so IWS will continue reading these values from the PLC addresses configured in the body of this driver worksheet.
   - **Enable Write on Tag Change** text box (*Boolean values, `1` = yes and `0` = no*): Type `1` so IWS will write the value from the tags configured in this driver worksheet whenever the tags change value.
   - **Station** text box (indicates the I/O device number to be accessed by this driver. Typically, the PLC is specified as Device #`1`.) Type `1`.
   - **Header** text box: Type a reference value from the following table.

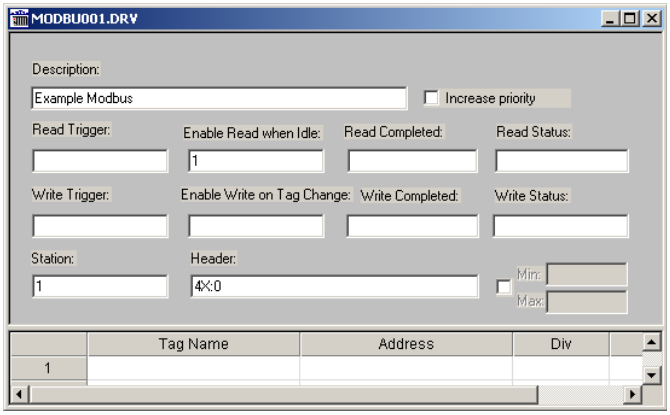
> **Note:**
> You must use a driver-specific format in the **Header** text box. The format for a ModBus protocol is:
> ***<reference for the type of register>:***
> ***<initial offset for the first stored value>.***

The following table provides a list of acceptable register types for the MODBU driver:

Register Type	Description

0X	Coil Status
1X	Input Status
3X	Input Register
4X	Holding Register
ID	Slave ID Number

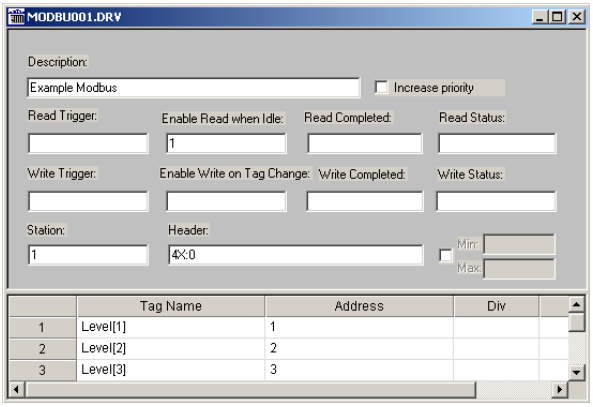
For example: **`4X: 0`** signifies a Holding Register with an offset of zero.



Sample Header Section

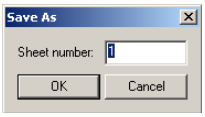
10. Type the following information into the spreadsheet provided at the body section of the *Modbu001.drv* window:
    - **Tag Name**: Type the I/O tag name.
    - **Address**: Type the PLC Address offset.
11. Associate the application tags to the PLC addresses according to the following table:

Application Tag	Driver Worksheet		PLC Address
	Header	Address	
Level[1]	4X:0	1	4X:1 (Holding Register 1)
Level[2]	4X:0	2	4X:2 (Holding Register 2)
Level[3]	4X:0	3	4X:3 (Holding Register 3)



Completed Body Section

12. Save the communication driver sheet (**File** → **Save**). When prompted to choose the driver sheet number, type `1` and click **OK** to save the *MODBU001.DRV* driver sheet.

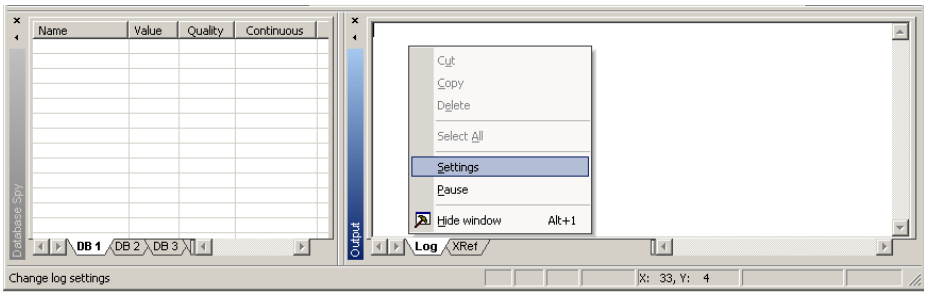


Save the Driver Sheet

To test the communication driver, use the following steps:

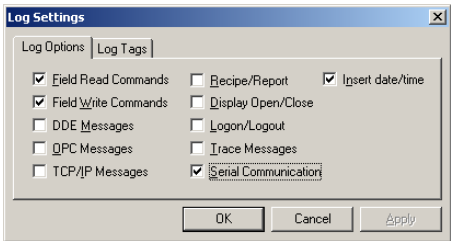
1. Click the **Run application** icon (on the IWS toolbar) to execute the application.

2. Right-click in the *Output* window (bottom of IWS development environment) and select **Settings** from the pop-up menu.



Select Settings from the Popup Menu

3. Enable the **Field Read Commands**, **Field Write Commands**, and **Serial Communication** options on the *Log Settings* dialog (see the following figure) then click **OK**.



Specifying the Log Options

4. Now you can start the application and monitor the communication messages from the *Output* window.

> **Note:**
> You can also monitor communication messages from the *LogWin* window (**Project** → **Settings** → **LogWin**).

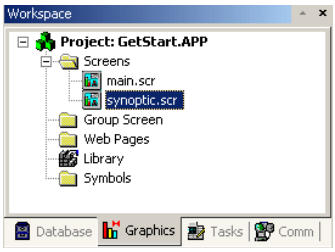
Chapter 7. Running the Web-Based Application

This chapter explains how to prepare your application to run on the Web.

> **Note:**
> IWS stores all application screens in the *Screens* folder, located on the *Graphics* tab in the *Workspace*.

To open your application screen, use the following steps:

1. Expand the *Screens* folder and double-click on **`synoptic.scr`**.

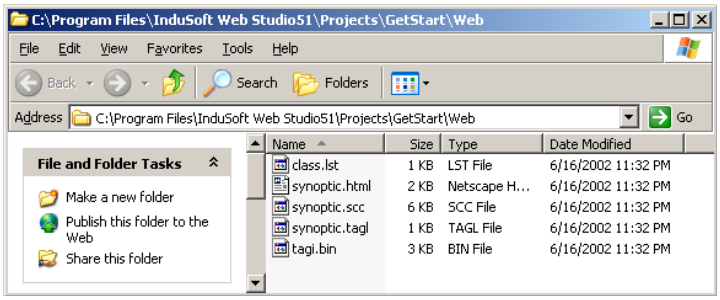


Select `scynoptic.scr`.

2. Select **File → Save All as HTML** to save all screens in HTML format.

Saving Screens in HTML Format

3. After executing this command, the Web files are stored in the `\Web` subfolder of the application directory.


C:\Program Files\InduSoft Web Studio51\Projects\GetStart\Web
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Program Files\InduSoft Web Studio51\Projects\GetStart\Web
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Name Size Type Date Modified
class.lst 1 KB LST File 6/16/2002 11:32 PM
synoptic.html 2 KB Netscape H... 6/16/2002 11:32 PM
synoptic.scc 6 KB SCC File 6/16/2002 11:32 PM
synoptic.tagl 1 KB TAGL File 6/16/2002 11:32 PM
tagi.bin 3 KB BIN File 6/16/2002 11:32 PM

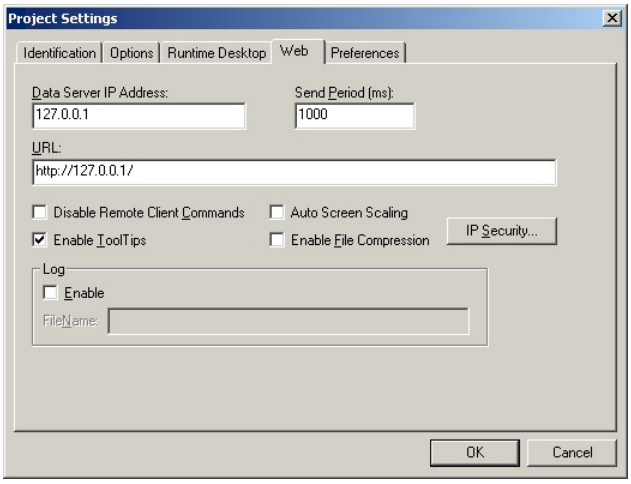
Viewing Web Files

4. Select **Project** → **Settings** from main menubar and select the *Web* tab.

5. You must specify the Server station IP address (for the station on which you are running IWS or CEView)the **Data Server IP Address** field, during runtime. The Web Thin Client station exchanges on-line data (tag values) with the station specified in this field.

6. You also must specify the URL path to the Web files (files saved in the *Web* subfolder) in the **URL** field. The URL depends on the Home directory configured for your server station Web Server.

   If the application's **`/Web`** subfolder is the home directory, you can configure the Web settings as shown in the following figure to test the Web Thin Client locally.



Specifying the Project Settings

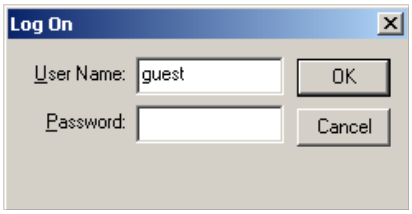
> **Note:**
> Microsoft provides Web servers for any Microsoft operating system. Consult your Microsoft documentation for information about installing and configuring a Web Server.

7. After configuring the Web settings, click **OK** to close the *Project Settings* dialog.
8. Close all screens in the screen editor (**Window** → **Close All**) and then select **Tools** → **Verify Application** to update the **Web Settings** for the Web pages.

> **Caution:**
> You must execute **Tools** → **Verify Application** after changing any settings in the **Project Settings** menu.

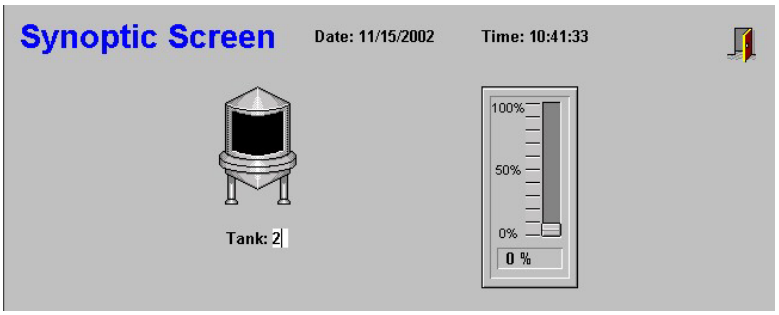
To test your Web-based application, use the following steps:

1. Click on **Run application** icon (on the IWS toolbar) to execute the application locally on the Server station.
2. Open an Internet Browser (Microsoft Internet Explorer or Netscape) and type the URL address to open the *synoptic.html* screen from the Server station (for example: *http://127.0.0.1/synoptic.html*).
3. When the *Log On* dialog displays in the Browser (see the following figure), type guest in the **User Name** field, then click **OK** to open the *synoptic.html* screen in the Browser.



Logging On as Guest

Notice that you can modify the level of any tank locally (Server Station) using the Viewer run-time module or remotely (Web Thin Client) using the Browser.



Modifying the Tank Level

Note:

A Web Thin Client requires an ActiveX component (*ISSymbol.ocx*) to handle screens on the Browser. If you connect the Web Thin Client to the Internet, this component is downloaded and registered automatically. Otherwise, you must copy the *ISSymbol.cab* from IWS's \BIN subfolder and paste it into the \<*OSPath*>*System32* directory on the Web Thin Client station. Use the WinZip® utility to unzip (*extract*) the files from *ISSymbol.cab* into the \<OSPath>\System32 directory and register the *ISSymbol.ocx* using the **regsvr32 ISSymbol.ocx** command.

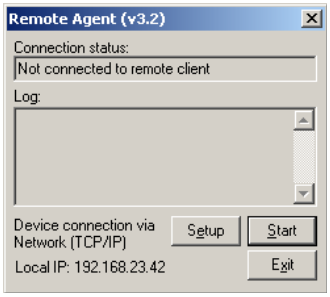
Chapter 8. Running Applications from a Remote Station

This chapter explains how to run your application from a remote location.

After configuring an application and testing it locally (on the development station), you can download it to a remote run-time station that is running IWS on Windows NT/2000/XP or CEView on Windows CE.

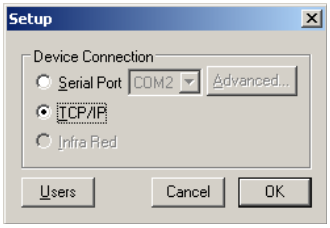
1. From the remote target station, select **Start** → **InduSoft Web Studio** → **InduSoft Remote Agent** to verify that the Remote Agent (**`CEServer.exe`**) is running.

2. When the *Remote Agent* dialog displays, click the **Setup** button.



Verifying the Remote Agent

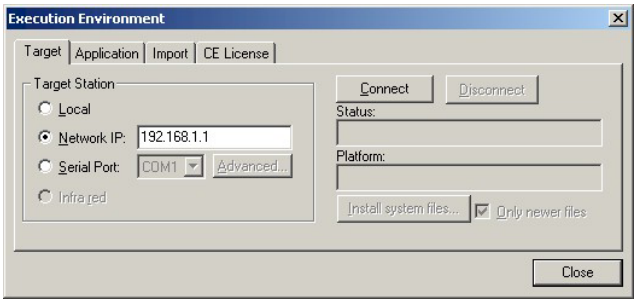
3. When the *Setup* dialog displays, click **Serial** or **TCP/IP** to specify how the device is connected to the development station.



Specifying the Device Connection Method

> **Note:**
> InduSoft recommends using TCP/IP for performance reasons.

4. Click **OK** and leave the Remote Agent running on the remote station.
5. Select **Project** → **Execution Environment** from the main menubar to open the *Execution Environment* dialog.
6. Specify a link type for the Target Station (**Network IP** or **Serial Port**). If you select **Network IP**, type the remote station IP Address into the text box.

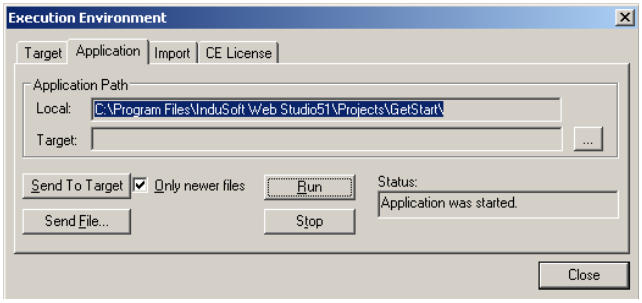


Specifying the Target Station Link Type

7. Click the **Connect** button to connect to the remote station.

> **Note:**
> If the remote station is a CE device, click the **Install system files** button to download the CEView runtime files to the remote station.

8. Return to the *Execution Environment* dialog and select the *Application* tab. Click **Send to Target** to download the application to the remote station.



Downloading the Application

9. After all of the application files are downloaded click **Run** to execute the application on the remote target station.

Revision History

Revision	Author	Date	Comments
A	Fabio Terezinho	May 09, 2002	Initial revision
B	Fabio Terezinho	June 17, 2002	Overall changes in the layout and content of the document
C	K. C. Francis	July 10, 2002	Overall changes in the layout and content of the document
D	K. C. Francis	November 13, 2002	Overall changes in the layout of the document
E	S. Warnke	November 17, 2003	New screen captures